

(19)日本國特許庁 (J P)

(12) 公表特許公報 (A)

(11)特許出願公表番号

特表平8-504977

(43)公表日 平成8年(1996)5月28日

(51) Int. Cl. 6 識別記号 廷内整理番号 F I  
G 0 6 F 9/38 3 5 0 A 7230-5 B  
3 7 0 C 7230-5 B

2014-2015 学年高二物理期中考试卷 (一) 答案及评分标准

(21)出願番号 特願平6-509063  
(86)(22)出願日 平成5年(1993)9月3日  
(85)翻訳文提出日 平成7年(1995)3月29日  
(86)国際出願番号 PCT/US93/08331  
(87)国際公開番号 WO94/08287  
(87)国際公開日 平成6年(1994)4月14日  
(31)優先権主張番号 07/954,084  
(32)優先日 1992年9月29日  
(33)優先権主張国 米国(US)  
(81)指定国 EP(AT, BE, CH, DE,  
DK, ES, FR, GB, GR, IE, IT, LU, M  
C, NL, PT, SE), JP, KR

(71)出願人 セイコーエプソン株式会社  
東京都新宿区西新宿2丁目4番1号

(72)発明者 センター シェリル  
アメリカ合衆国 カリフォルニア州 9501  
4 クパチーノ, セレスティ サークル 2074  
6

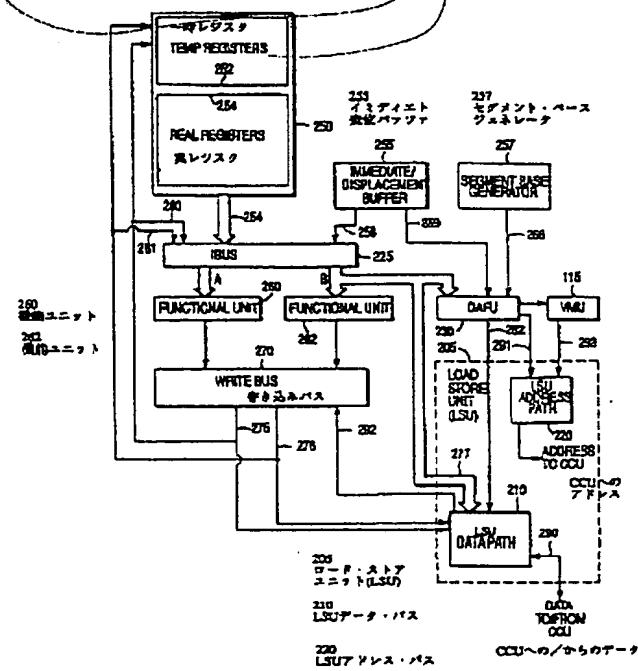
(72)発明者 ワング ジョハネス  
アメリカ合衆国 カリフォルニア州 9406  
2 レッドウッド シティ、キングストリ  
ート 25

(74)代理人 弁理士 鈴木 喜三郎 (外1名)

(54)【発明の名称】スーパースカラ・マイクロプロセサにおけるロード及び／又はストア動作を扱うシステム及び方法

(57) 【要約】

本発明はRISC型スパースカラ・アーキテクチャ環境について、メモリからの読み出し及びメモリへの書き込み或いは入出力に必要な、ロード及びストア・オペレーションを管理するためのシステム及び方法を提供するものである。このタスクを行うためにロード・ストア・ユニットが設置され、ロード・ストア・ユニットの主な目的は可能な限りロード要求をアウト・オブ・オーダで行い、ロード・データを命令実行ユニットが使用できるように最大速度で返すことである。ロード・オペレーションはアドレス衝突も書き込み実行待ちも存在しない時にのみアウト・オブ・オーダで実行できる。アドレス衝突が発生するのは、先行の命令が書き込み中の記憶場所に於いて読み出しが要求された時である。書き込み実行待ちとは先行の命令がストア・オペレーションを要求したが、ストア・アドレスが未だ計算されていない状態をいう。データ・キャッシュ・ユニットは8バイトの位置合わせされていないデータを返す。このデータが命令実行ユニットに戻される前に、ロード/ストア・ユニットがこのデータを正しく位置合わせする。従って、ロード・スト



## 【特許請求の範囲】

特許請求の範囲は以下の通りである。

1. プログラム・ストリームを実行するマイクロプロセサ・システムで、前記マイクロプロセサ・システムは、

(a)命令ストアから命令を取りだして予め決められた複数の前記命令を命令バッファに供給する命令フェッチ・ユニットと、

(b)前記命令フェッチ・ユニットに接続された、前記命令バッファからの前記複数の前記命令をアウト・オブ・オーダで実行する実行ユニットで、前記実行ユニットはロード要求をメモリ・システムに対してアウト・オブ・オーダでストア要求をイン・オーダで行なうように選択性化されたロード・ストア・ユニットを備え、前記ロード・ストア・ユニットは、

(i) 実行中の前記複数の前記命令に対応する複数のアドレスを管理するように選択性化されたアドレス・バスと、

(ii)アドレス衝突及び書き込み実行待ちが実行中前記複数の前記命令の各々の間に存在するか否かを検出して信号を送るアドレス衝突手段で、アドレス衝突も書き込み実行待ちも検出されなければ前記ロード・ストア・ユニットが前記ロード要求を実行する、アドレス衝突手段と、

(iii)ロード及び/又はストア・データを前記メモリ・システム及び前記実行ユニット間で伝送するデータ・バスで、前記データ・バスは前記メモリ・システムから返されたデータを位置合わせし、それによりワード境界と一致するデータが前記メモリから正しいアライメントで前記実行ユニットに返されるように構成されたデータ・バスと、

を有する、実行ユニットと、

から成ることを特徴とするマイクロプロセサ・システム。

2. 前記アドレス・バスは前記ロード及び/又は前記ストア要求の上位バイト及び下位バイトを格納するための複数のアドレス・バッファを含むことを特徴とする請求項1に記載のシステム。

3. 前記データがワード境界と一致する場合、前記ロード・ストア・ユニット

が複数のメモリ要求をメモリに対して行なう手段を含むことを特徴とする請求項1に記載のシステム。

4. 前記コード・ストア・ユニットに接続され、前記複数の命令用のアドレスを計算するために選択性化されたデータ・アドレス機能ユニットからさらに成ることを特徴とする請求項1に記載のシステム。

5. 假想アドレスから生成される物理アドレス変換を前記実行ユニット及び前記コード・ストア・ユニットに供給されるように選択性化された假想メモリユニットからさらに成ることを特徴とする請求項4に記載のシステム。

6. 前記ロード・ストア・ユニットはメモリ要求を行なう前に前記データ・アドレス機能及び前記假想メモリユニットからの物理アドレスを必要とすることを特徴とする請求項5に記載のシステム。

7. 前記命令がCISC型命令であり、そして前記命令実行ユニットが前記CISC型命令をRISC型命令にデコードするデコード手段から成ることを特徴とする請求項1に記載のシステム。

8. 前記コード・ストア・ユニットがメモリから受け取るデータを行き先レジスタの元々の内容とマージさせる手段から成ることを特徴とする請求項1に記載のシステム。

9. 前記データ・バスにロード及び/又は実行データを直接転送し、それによって後続のストア・オペレーションがその後に直ちに実行されるようにするために前記実行ユニット及び前記ロード・ストア・データ・バス間にデータ線を備えることを特徴とする請求項1に記載のシステム。

10. 前記命令バッファ中の命令の比較的新しさを示す履歴ポインタからさらに成ることを特徴とする請求項1に記載のシステム。

11. 前記衝突手段が、アドレス衝突又は実行待ちストア・アドレスが存在するかどうかを決定することによってロード依存性を示すことを特徴とする請求項1に記載のシステム。

12. システムの状態を不正に変更するようなロード・バイパスを防止する手段からさらに成ることを特徴とする請求項1に記載のシステム。

13. 前記ロード・ストア・データ・バスにストア・データを直接供給するためには假想ユニットの結果を監視する(スヌーピング)手段からさらに成ることを特徴とする請求項1に記載のシステム。

14. 前記ロード・ストア・ユニットが浮動小数点演算用に別個のロード・ストア・データ・バスを含むことを特徴とする請求項1に記載のシステム。

15. 前記実行ユニットにはレジスタ・ファイルが含まれ、レジスタ・ファイルには複数の実バッファ及び複数の一時バッファが含まれることを特徴とする請求項1に記載のシステム。

16. コード命令をアウト・オブ・オーダで発行する実行ユニットを有するRISC型スーパースカラ・マイクロプロセサに於いて、メモリ素子との間のロード並びにストア要求を管理する方法で、その方法は、

(1) 命令ウインドウから選択された命令のアドレスを計算し、前記アドレスを一つのロード・ストア・ユニットに転送するステップと、

(2) 前記命令がロード・オペレーション、ストア・オペレーション、実行オペレーション、或いは前記コード、前記ストア、及び前記実行の各オペレーションの組み合わせを伴うか否かを決定するステップと、

(3) 前記命令にロード・オペレーションが含まれる場合、アドレス衝突及び書き込み実行待ちが存在するか否かを検査し、そして前記検査の結果を知らせるステップと、

(4) 优先度スキーム及び前記検査ステップ(3)の結果に基づき前記メモリ素子に要求を行なうステップと、

(5) 要求されたデータを前記ロード・オペレーション及び/又は前記ストア・オペレーションから前記ロード・ストア・ユニットのデータ・バス部分に於いて受け取るステップと、

(6) 前記要求データが位置合わせされていないデータである場合、前記要求データの位置合わせを行なうステップと、

から成ることを特徴とするコード及びストア要求を管理する方法。

17. 発行のステップ(1)は前記命令に対してデータ依存性検査を行なうステップが含むことを特徴とする請求項16に記載の方法。

18. 前記命令の結果を一時バッファ内の、予め決められた場所に書き込むステップからさらに成ることを特徴とする請求項16に記載の方法。

19. 前記一時バッファをバイパスすることによって前記コード・ストア・ユニットにデータを供給するステップからさらに成ることを特徴とする請求項18に記載の方法。

20. 全てのストア・オペレーション要求をプログラムの順序に行なうステップからさらに成ることを特徴とする請求項16に記載の方法。

21. システムの状態を不正に変更するロード・オペレーションのロード・バイパスを防止するステップからさらに成ることを特徴とする請求項16に記載の方法。

22. メモリから受け取ったデータを行き先レジスタの元々の内容とマージさせるステップからさらに成ることを特徴とする請求項16に記載の方法。

## 【発明の詳細な説明】

スーパースカラ・マイクロプロセサにおけるロード及び/又はストア動作を扱うシステム及び方法  
(SYSTEM AND METHOD FOR HANDLING LOAD AND/OR STORE OPERATIONS IN A SUPERSCALAR MICROPROCESSOR)

発明者： シェリルD. センター (Cheryl D. Senter)  
ヨハネス・ワング (Johannes Wang)

関連特許出願の引継

本出願は本出願の代理人に譲渡されている下記の出願に関連するものである。すなわち、ニューエン (Nguyen) その他による1991年7月8日出願の米国特許出願番号 07/727,058 (代理人登録番号SP021)、「拡張可能RISCマイクロプロセサ・アーキテクチャ」(EXTENSIBLE RISC MICROPROCESSOR ARCHITECTURE)、及び 058出願の絶続出願である、1992年1月8日出願の出願番号07/817,809に関連する。上記出願の開示を参照することによって当該特許出願の明細書の記載内容が本明細書に組み込まれているものとする。

## 発明の背景

## 1. 産業上の利用分野

本発明は一般的にはスーパースカラ・マイクロプロセサの設計に適し、より具体的には、命令をアウト・オブ・オーダで実行するマイクロプロセサに於けるロード及びストア動作を扱うシステム並びに方法に関する。

## 2. 関連技術

スーパースカラ縮小命令セット・コンピュータ (RISC) に於ける大きな課題は如何にして斯かる命令実行の本質的な問題である依存性によるデータ・エラーの発生を回避しつつ、複数の命令を平行処理的に、アウト・オブ・オーダで実行できるか、という点に関する。RISCプロセサに於ける最も簡単な命令発行方針は命令をプログラムでの順序と同じ順序で発行し (イン・オーダ発行)、結果をまた同じ順序で書き出すことである (イン・オーダ完了)。アウト・オブ・オーダ完了はイン・オーダ完了に比して複雑であるが、同じ複雑の動作に対してスーパース

に於けるデータ依存性のみを検査すればよいから、結果は当然正しい順序で書き出される。アウト・オブ・オーダ完了に於いてはまた、機能ユニットは結果バス及びレジスタ、ファイル書き込みポート間での調停を行なわなければならない。その理由は、同時に完了する全ての命令の必要を満たすために十分な数のバス及びポートが存在しないであろうからである。

更に、アウト・オブ・オーダの完了では命令例外の処理がより困難になる。ある条件下で命令が例外を生成した場合、その命令はハードウェアだけでは正しく実行できない。

デコードされた命令が資源競合を生じた場合、真の依存性を有する場合、あるいは未完了の命令に対して出力依存性を有する場合、イン・オーダ命令発行プロセサは命令のデコードを中止する。従って、後続する一つ又は複数の命令が実行可能であっても、プロセサは競合又は依存性を生じた命令の後に来るものを先読み (lookahead) 处理できない。従来の解決策はデコードを実行段階から分離して、命令が直ちに実行可能であるか否かにかかわらず命令のデコードを継続して行なえるようにすることである。この分離はデコード段階と命令段階の間に「命令ウインドウ」と呼ばれるバッファを配置することによって実施される。

先読みのために、プロセサは命令をデコードし、ウインドウ中に場所がある限りデコードされた命令を命令ウインドウに入れ、それと同時に、実行可能な命令 (すなわち、資源競合又は依存性を持たない命令) を見出すためにウインドウ中の命令を検査する。命令ウインドウは命令のプールとしての機能があり、この機能によってプロセサは先読みの能力を得る。この能力はウインドウのサイズとプロセサの命令フェッチ・ユニット (IFU) の性能によって制約されるものである。これによって、各命令の最初のプログラム順序と無関係にウインドウから発行できるので、命令のアウト・オブ・オーダ発行が可能となる。この場合、命令発行元に対する唯一の制約はプログラムが正常に動作することを保証するに必要な制約である。

如何なる特定の命令においても、発行に関する制約はイン・オーダ発行の場合と殆ど同じである。すなわち、命令は資源競合又は依存性を持たない場合発行され

カラ・プロセサの性能向上に効果がある。例えば、アウト・オブ・オーダ完了はロード又は浮動小数点演算等の長い命令時間動作の性能改善のために使用される。機能ユニット内で実行中の命令の数に制限はないが、その最大数は全ての機能ユニット内のバイオペラント段階の数である。この場合、命令はアウト・オブ・オーダで完了でき、その理由は結果の計算にひとつの機能ユニットが1サイクル以上のサイクルを費やした場合でも命令の発行に停止がないからである。従って、後続の命令が終了した後、機能ユニットが一つの命令を完了しても構わない。

下記のコードシーケンスを考えて見る。ただし、'op' はオペレーション、'Rn' は番号つきレジスタ、' := ' は代入を表す。

R3 := R3 op R5 (1)  
R4 := R3 + 1 (2)  
R3 := R5 + 1 (3)  
R7 := R3 op R4 (4)

この場合、一般的にアウト・オブ・オーダの命令完了が可能であっても、第1命令の代入は第3命令の代入の後には完了できない。第1命令と第3命令がアウト・オブ・オーダで実行された場合、異常且つ不正な値がR3レジスタに残り、例えば第4命令が不正なオペレランド値を受け取るような事態が生じる。第3命令の結果は第1命令に対して「出力依存性」を有し、このコードシーケンスにおいて正しい出力値を得るためにには第3命令は第1命令の後に完了しなければならない。従って、第3命令の結果が、計算にもっと時間の掛かるより古い命令によって上書きされる場合、第3命令の発行は待たなければならない。

アウト・オブ・オーダの完了は性能の向上をもたらすものであるが、より多くのハードウェア、つまりデータ依存性論理を必要とする。アウト・オブ・オーダの完了の場合、データ依存性論理は複雑になる。その理由は、この論理ではデコードされた命令と、全てのバイオペラント段階の全ての命令の間に於けるデータ依存性の検査が行なわれるからである。結果が正しい順序で書き出されることを保証するのもハードウェアの役割である。これに対して、イン・オーダ実行に於いては、データ依存性論理はデコードされた命令と、現在実行中の幾つかの命令の

るのである。アウト・オブ・オーダの発行によって、プロセサは発行可能なより大きな命令集合を得ることになり、それによって、同時に実行可能な命令をプロセサが見つけ得る確率が高まる。しかし、命令をアウト・オブ・オーダで発行する能力によって他の発行制約が生じる。これは命令をアウト・オブ・オーダで実行する能力に於いて出力依存性の制約が導入されるに類似している。

これを理解するために上記のコードシーケンスの例を振り返って見る。第2命令の実行が始まる前には第3命令の代入は完了できない。さもなければ、第3命令が第2命令の第1オペラントを不正に上書きすることがあり得る。第3命令の結果は第2命令の第1入力オペラントに対して「反依存性」を有するといわれる。「反依存性」という用語は、反依存性制約はそれが逆になった以外には其の依存性に関する制約と同様であることを意味する。第2命令が使用する値を第1命令が生成するかわりに、第2命令は第1命令が使用する値を後続する値を生成する。これを防止するためには、第2命令が始まるまでプロセサは第3命令を発行してはならない。第2命令は第1命令に依存するから、第3命令は他の点では独立していても第1命令が完了するまで待たなければならない。

反依存性が重要なのは主に命令がアウト・オブ・オーダで発行可能な場合である。正常なオペレーション中、停止した命令の入力オペラントは後続する命令によって破壊されることがある。しかし、スカラ・プロセサにおいては、往々にして命令例外は例外条件を修正し、そして問題を生じた命令を再試行することによって処理される。この命令がアウト・オブ・オーダで完了した場合、その命令の再試行が行なわれた時、後続する命令によってその再試行中の命令の入力オペラントが上書きされることがあり得る。この問題は正確な割り込みをサポートするプロセサでは起こり得ない。この問題の解決には、再起動を可能にするためにプロセサが命令オペラントのコピーを維持する必要があるかもしれない。

プログラム命令によって行われるオペレーションの代表的な二つ

のオペレーションはロード及びストアのオペレーションである。一般的に、ロー

ド及びストア・オペレーションはそれぞれ記憶場所を読み出し、変更する。他のプログラム命令と同様に、ロード及びストアはアウト・オブ・オーダで実行できる。ロード及びストアは同時にデコード可能であるが、従来的には1サイクル当たり一つのロード又はストアのみが発行される。データ・キャッシュの使用においては、通常、ロードはストアに優先される。その理由は多くの場合ロードはプロセサが演算を行うのに必要な値を生成するからである。データ・キャッシュの使用において、ストアがロードと競合する場合、ストアの実行が可能になるまで、通常ストアはストア・バッファに保持される。更に、従来的には、ストアは他のストアに対してプログラム順序で実行され、ロードも含めて全ての他の先行の命令が実行された後にのみ実行される。これによって、データ・キャッシュ使用においてのプロセサのイン・オーダ状態が保存される。その理由はキャッシュの更新はそれが絶対に正しく行なわれ得るまで行なわれないからである。ストア・バッファの使用によってストアが正しい順序で保持され、先行の命令が完了するまでストアの完了が延期されるのである。

ストアは他の先行の命令が実行されるまで保持され、そしてロードはプロセサ中の計算に必要な値を生成するから、ストアに対してロードをプログラム順序に保持することは性能に対して重大な悪影響を及ぼす。全ての先行のストアが完了するまでロードが待たなければならない場合、そしてそのために最も新しいストアに先行する全ての命令が完了するまでロードが待つ場合、ロード・データに依存した、ロードに後続する全ての命令も待つことになる。この性能上の問題を回避するために、ロードはストア・バッファで待機している、先行のストアをバイパスすることができ、ロード・データは後続の計算において使用できる。

ロードが先行のストアをバイパスすることができる場合、ロードは未だ実行されていない先行のストアからデータを得る必要があるかも知れない。プロセサはロードが先行のストアに対して有する直の依存性を、ロードの仮想記憶アドレスと、全ての未完了の先行ストアの仮想記憶アドレスとを比較することによって検査する（仮想アドレスとは記憶管理ユニットによるアドレス変換が適用される前に、命令によって直接計算されるアドレスである）。ここに於いて、各仮想アドレ

スに対して一意的なマッピングの存在が仮定される。その理由は2個の異なる仮想アドレスが同じ物理的なアドレスへアクセスすることができないようにするためにである。この仮定に基づき、仮想アドレス比較は物理的記憶場所間の全ての依存性を検出する。ロード・アドレスが先行のストアのアドレスと一致する場合、又は先行のストアのアドレスのいずれも未だ計算されていない場合（この場合、依存性は検出不可能なので、依存性の存在が仮定される）、一つのロードは一つのストアに対して真の依存性を有する。ロードがストアに依存する場合、データ・キャッシュは正しい値を持たないので、そのロードはデータ・キャッシュによって演算されない。ストアの有効アドレスが後続するロードのアドレスと一致する場合、そのロードはストアが完了するのを待つ代わりに、ストア・データが有効な場合、ストア・バッファによって直接演算される。

上述の如く、ロード及びストアは記憶場所に対する反依存性及び出力依存性を回避するような方法で実行される。ロードは先行のストアをバイパスできるが、ストアは先行のロードをバイパスできない。従って、ロードとストアの間には反依存性は存在しない。一つのストアは従来他のストアに対して通常プログラム順に発行されるので、ストア間には出力依存性は存在しない。

従来的には、データ・キャッシュに於いてコードは他のロードに

対してプログラム順に実行される。当事者の意見ではロードをアウト・オブ・オーダで実行することにより性能の点で得られる利点はないとのことである。その理由は、古いロードによってプロセサに供給されるデータは新しいロードによって供給されるデータよりも計算に必要とされることが多いからである。

上記の概念の詳細は幾つかの刊行物で論じられている。例えば、John L. Hennessy その他著、「Computer Architecture - A Quantitative Approach」（Morgan Kaufmann Publishers, Inc., San Mateo, California, 1990年発行）及びMike Johnson著「Superscalar Microprocessor Design」（Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1991年発行）（特に第8章、この章の一部は上に転載されている）。両者とも参照することによって全文が本明細書に組み込まれているものとする。

### 発明の要約

本発明はスーパースカラ3ISC型マイクロプロセサ・アーキテクチャ環境においてメモリからの読み出し及びメモリへの書き込み或いは入出力に必要なロード及びストア・オペレーションを管理するシステムを提供するものである。本発明はプログラム・ストリームを実行するマイクロプロセサ・システムを提供するもので、このシステムには命令を命令ストアより取り出し、且つ予め決められた複数の命令を命令バッファに供給する命令フェッチャ・ユニットが含まれている。更に、命令フェッチャ・ユニットと結合している実行ユニットが含まれ、実行ユニットは命令バッファからの複数の命令をアウト・オブ・オーダで実行するためにある。実行ユニットにはロード・ストア・ユニットが含まれ、ユニット

はアウト・オブ・オーダなロード要求とイン・オーダのストア要求をメモリ・システムに対して行なうように適応化されている。従って、本発明のロード/ストア・ユニットの主な目的には、可能な限り、アウト・オブ・オーダなロード要求を行ない、なるべく速やかにロード・データを命令実行ユニットに返すことである。コード・オペレーションはアドレス衝突がなく、実行待ちの書き込みオペレーションが存在しない時のアウト・オブ・オーダで実行できる。アドレス衝突が発生するのに、古い命令がこれから書き込まれる記憶場所に於いて読み出しが要求された時である。実行待ちの書き込みオペレーションとは、古い命令がストア・オペレーションを要求したがストア・アドレスの計算がまだ行なわれていないことである。データ・キャッシュ・ユニットは位置合わせされていない8バイトのデータを返す。ロード/ストア・ユニットはデータが命令実行ユニット（IEU）に返される前にデータの正しい位置合わせを行う。従って、ロード/ストア・バッファの三つの主要なクスクスは(1)アウト・オブ・オーダのキャッシュ要求の処理、(2)アドレス衝突の検出、及び(3)データの位置合わせである。

ロード・ストア・ユニットには現在実行中の複数の命令に対応する複数のアドレスを管理するために適応化されたアドレス・バス、現在実行中の複数の命令の各命令間にアドレス衝突及び実行待ちの書き込みオペレーションが存在するかどうかを検出して知らせるアドレス衝突手段が含まれ、そうすることによって、ロー

ド・ストア・ユニットはアドレス衝突も実行待ちの書き込みオペレーションも検出されなかった場合、ロード要求を実行する。ロード・ストア・ユニットは更にデータ・バスで構成され、データ・バスはロード及び1又はストア・データをメモリ・システム及び命令実行ユニットの間で転送する。データ・バスは記憶システムより返されたデータの位置合わせをし、斯くしてキャッシュの4ワード境界と一致しないデータがメモリ・システムから命令実行ユニットに正しいアライメントで返されるようとする。

### 図面の簡単な説明

本発明は添付の特許請求の範囲に具体的に掲示されている。本発明の上記の、そして後述の利点の理解を深めるために、次に図面を参照して説明する。すなわち、

第1図は本発明が動作するマイクロプロセサ・アーキテクチャ100のブロック図である。

第2図はロード・ストア・ユニット（LSU）205を含む命令実行ユニット（IEU）107を示す一般的なブロック図である。

第3図はLSUアドレス・バス220を示すブロック図である。

第4図はLSU205に位置するアドレス衝突ブロックを示す概略図である。

第5図はLSUデータ・バス210を示す概略図である。

第6図はキャッシュ線が交差する、位置合わせされていない複数ロードの一列を示す図である。

第7(a)図～第7(h)図はLSU205の動作の一例を示す図である。

### 発明の詳細な説明

第1図において、本発明の好適な実施例に基づいて、一般的に100で表わされるマイクロプロセサ・アーキテクチャが配置されている。システム・アーキテクチャ100にはホスト・プロセサ105、キャッシュ制御ユニット及びメモリ（CCU）110、仮想メモリ・ユニット（VNU）115、入出力ナブシステム180、メモリ制御及びインターフェース・ユニット120、及びインクリーブ・オペレーション用に構成さ

れたインターブド・メモリ・バンク (160a, 160b, 160c) (以降、主メモリ 160 と称す) が含まれている。主メモリ 160 は外部データバス 162 を介して MCUI20 に接続されている。本発明はマルチプロセッサ環境で動作すると予想されるので、その場合、他のプロセッサもメモリバス 162 に接続される。

ホストプロセッサ 105 は主メモリ 160 において各アドレス又は記憶場所に格納されているソフトウェア命令を実行する。これらのソフトウェア命令はホスト・プロセッサ 105 にプログラム・カウンタの制御の下にイン・オーダで転送される。しばしば、命令のうちあるものはホストプロセッサ 105 が一つ又は複数の周辺入出力接続 35 をアクセスすることを必要とする。

MCUI20 は一つの回路であり、この回路によってデータ及び命令は CCU110 (D\_キャッシュ 118 と I\_キャッシュ 119 (読み出し専用))、10U150、及び主メモリ 160 の間を転送される (読み出されるか書き込まれる)。MCUI20 にはスイッチ・ネットワーク 145 が含まれ、それにはスイッチ・アービトリエーション・ユニット 132、データ・キャッシュ・インターフェース・ユニット 117、命令キャッシュ・インターフェース回路 112、I/Oインターフェース回路 155、及びポートとして知られる一つ又は複数のメモリポート・インターフェース回路 148 が含まれる。各ポート・インターフェース

回路 148 にはポート・アービトリエーション・ユニット 134 が含まれている。

スイッチ・ネットワーク 145 はマスター装置とスレーブ装置の通信の手段である。スイッチ・ネットワーク 120 に対するマスター装置になり得るのは D\_キャッシュ 119 と I\_キャッシュ 118、又は I/O コントローラ・ユニット (10U) 150 である。スレーブ装置として機能し得るものは、例えば、メモリ・ポート 148 又は 10U150 である。スイッチ・ネットワーク 145 の機能は CCU110 (I\_キャッシュ 118 及び D\_キャッシュ 119) と 10U150 から様々な命令及びデータ要求を受け取ることである。これらのユニットをバス要求者と呼ぶ。これらの要求を受け取った後、スイッチ・アービトリエーション・ユニット 132 及びポート・アービトリエーション・ユニット 134 は要求を優先度に応じて並べ、適切なメモリポートに渡す (命令アドレスによる)。ポート 148、或いは場合によっては複数のポート、は次に必要なタ

タイミング信号を生成し、データを外部バス 162 に送り、あるいはデータを外部バス 162 から受け取る。

命令 フェッチ・ユニット (IFU) 106 及び命令実行ユニット (IEU) 107 はホスト・プロセッサ 105 の主要な動作機能である。IFU106 及び IEU107 の機能を直接サポートするために YMUI15、CCU110、及び MCUI20 が配置されている。IFU106 の主要な機能は命令の取り出し、IEU107 による実行を待つ命令のバッファリング、そして、一般的に、次の命令の取り出しに使用される次の仮想アドレスの計算、である。各命令は命令バス 101 を介して I\_キャッシュ 119 から IFU106 によって同時に取り出される。命令は「パケット」或いは 4 個の命令の集合に入れられる。命令集合の転送は制御線 102 を介して供給される制御信号によって、IFU106 及び CCU110 の間で調整される。取り出される命令の仮

想アドレスは IFU 制御及びアドレス・バス 103 を介して IFU106 によって YMUI15 に供給される。YMUI15 へのアクセスに従事する調停の必要性は、IFU106 及び IEU107 の両者とも YMUI15 を共通の、共有の資源として使用することから生じる。アーキテクチャ 100 の好適な実施例において、仮想アドレスの物理ページ内のアドレスを定義する下位ビットは IFU106 によって、制御線 102 を介して CCU110 に直接転送される。IFU106 によって供給される仮想アドレスの仮想化上位ビットはバス 103、104 のアドレス部分によって YMUI15 に供給され、そこで対応する物理ページ・アドレスに変換される。IFU106 にとっては、この物理アドレスは変換要求が YMUI15 に出された 1/2 内部プロセッサ・サイクル後、制御線 111 を介して YMUI15 から CCU110 に直接転送される。

一方、IFU106 によって取り出された命令ストリームは命令ストリーム・バス 108 を介して IEU107 に供給される。制御信号は制御線 109 を介して IFU106 と IEU107 の間でやり取りされる。

IEU107 はデータを双方向データ・バス 112 を介して D\_キャッシュ 215 に格納し、またそれから検索する。IEU107 によるデータ・アクセスの場合、物理アドレス全体が制御バス 113 のアドレス部分によって CCU110 に供給される。IEU107 は YMUI15 を、仮想データ・アドレスを、CCU110 への送り出しに適切な物理データ・アドレ

スに変換する資源として利用する。IFU106 に対するオペレーションとは異なり、YMUI15 は対応する物理アドレスをバス 104 を介して IEU107 に返す。

CCU110 はホスト・プロセッサ 105 と主メモリ 160 との間のバッファとして使用される。一般的に、CCU110 は小型の、高遅メモリで、ホスト・プロセッサ 105 の近傍に位置し、最も最近アクセスされたコード又はデータを保持する。CCU110 は、適切であれば物理ア

ドレスで定義されたデータ要求が命令及びデータ・キャッシュ 118、119 から読み出されるか否かを決定するという一般的に従来的な高レベルの機能を行なう。命令キャッシュあるいはデータ・キャッシュ 118、119 へのアクセスによってアクセス要求が満たせる場合、CCU110 はデータ・バス 101、113 を通じてデータ転送を調整し実行する。命令キャッシュあるいはデータ・キャッシュ 118、119 へのアクセスによってアクセス要求が満たせない場合、CCU110 は対応する物理アドレスを MCUI20 に供給する。この場合、物理アドレスの他に、主メモリ 160 への読み出し又は書き込みアクセスが必要であるかを識別するために十分な制御情報、各要求のソース又は行き先キャッシュ 118、119、又は要求されたオペレーションが IFU106 又は IEU107 によって発行された最終的なデータ要求と関連付けられるための追加の識別情報が CCU110 によって MCUI20 に供給される。

第 2 図に IEU107 のデータ・バスの代表的な高レベルのブロック図を示す。IEU107 の目的は最小限の時間で最大数の命令を実行することである。IEU107 にはレジスタ・ファイル 250、ロード/ストア・ユニット (LSU) 205、命令バス (IBUS) 225、一式の機能ユニット 260、262、230、イミディエイト変位バッファ 255、セグメント・ベース・ジェネレータ 257、及び書き込みバス 270 が含まれている。LSU205 は LSU アドレス・バス 220 及び LSU データ・バス 210 の二つの部分に分かれている。

スーパースカラ制御ブロック (図示せず) はデータ依存性検査を行い、必要な機能ユニット 260、262、及び 230 が使用可能であるかどうかを検査することによって、ある命令が発行可能であるか否かを決定する。一旦スーパースカラ制御ブロックが一つの命令を発行するように決定すると、IBUS225 は発行される命令が必要とするデータを検索する (検索はレジスタ・ファイル 250) バ

イバス・データ 280、282、或いはイミディエイト・データ 258、259 から行われる)。IBUS225 は複数のマルチプレクサによって構成され、これらのマルチプレクサが、どのデータが機能ユニット 260、262、230 に転送されるかを選択する。IBUS225 は A バスと B バスと呼ばれる一对のバスにデータの転送を行なう。選択されたデータは、機能ユニット 260、262、230 のうちどの機能ユニットがその命令によって使用されるか、或いは現在実行中の命令のオペレーションによって必要とされているか、を決定することによって、A バスか B バスのどちらかのバスに入れられる。

ほとんどの命令の入力及び出力は複数のレジスタ・ファイルのうち一つのレジスタ・ファイルから送られる、つまり格納されている。好適な実施例では、各レジスタ・ファイル 250 (例えば、別個の整数、浮動小数点、或いはブルー・レジスタ・ファイル) は 32 個の実数エントリ 254 及び 8 個の一時バッファ 252 のグループを有する。一つの命令が完了すると (「完了」とはオペレーションが終了し、オペランドはその行き先レジスタに書き込める状態にあることをいう)、その結果は一時バッファ 252 中の事前に割り当てられた場所に格納される。これらの結果は後に実レジスタ 254 中の適切な場所に書き込まれる。このような結果の一時バッファ 252 から実レジスタ 254 への移動は「退避」 (retirement) と呼ばれる。一度に複数の命令が退避できる。退避により、コンピュータのプログラムカウンタを含めて、マシンの「公式な状態」の更新が行なわれる。

命令は「パケット」と呼ばれる 4 個のグループ毎に命令コード FFIPO (first-in-first-out) 先入れ先出し方式) レジスタ・スタック記憶装置 (図示せず) (本明細書では命令ウィンドウと呼ぶ) を介して IFU106 から IEU107 へ送られる。パケットはロード、ストア、及び 2 個の実行ユニットで構成される 4 個のユニットに分

解される。パケットがこれら 4 個のユニットに分解された理由はシステム 100 はロード、ストア、実行の各オペレーション又はそれら全ての組み合わせを実行できる命令を使用して動作するからである。従って、本発明はこれら三つの場合の全てを処理できるパケットを供給するものである。

IEU107は一度に4個のパケットの命令までデコードしスケジュールできる。命令ウィンドウは全部で16個の命令を4個のパケットに格納する。IEU107は命令ウィンドウを検査し、各サイクルごとにIEU107は命令ウィンドウから最大数の命令を発行しようとする。一旦1個のパケット中の全ての命令が実行され、それらの結果がプロセサのレジスタ・ファイル250に格納されると、そのパケットは命令ウィンドウからフラッシュされ、次に新しいパケットが命令ウィンドウに格納される。

一旦その命令が発行されると、レジスタ・ファイル250中の諸レジスタがアクセス可能となる。一時レジスタ252は先行の命令によって生成されたデータに対してデータ依存性を持っていた命令が実行されるとアクセスされる。レジスタ・ファイル250からのデータはデータ線254を介してLSU225に転送される。

DAFU230はLSU205によって使用される32ビットのリニア・アドレスを計算する。DAFU230では複数の異なるアドレス指定モードがサポートされている。2サイクルを必要とするデータが4ワード境界を越える場合、そのデータの最初と最後のアドレスはDAFU230によって計算される。アドレスを形成するために4個までのコンポーネントが加算される。すなわち、セグメント・ベース、ベース・レジスタ、スケールド・インデックス・レジスタ、及び変位値、の4個のコンポーネントである。セグメント・ベースには目的のメモリ・セグメントの開始アドレスが含まれている。

ベース及びインデックス・レジスタはレジスタ・ファイル250中のどの32ビットレジスタであっても構わない。インデックス・レジスタはそれを1、2、4、又は8で乗算することによってスケールされる。変位値は命令中に存在する定数値（イミディエート値）である。これらのフィールドのうちどのフィールドも省略可能であり、新しくしてアドレス演算に於ける最大限の自由度が得られる。

セグメント・ベースはセグメント・レジスタ・ブロック257から得られる。セグメント・ベース・ジェネレータ257はデータが如何にしてメモリ中で分割されているかを示す一つの値を生成し、この値をデータ線266を介してDAFU230に転送する。変位はイミディエート変位バッファ255から得られる。イミディエート変位

データ・バス210は書き込みバス270、CCU110、DAFU230、及びIBUS225とインタフェースする。LSUの三つの主要なタスクは(1)アウト・オブ・オーダで実行される。ロード要求を伴う場合、LSUデータ・バス210に直接供給される。ロード及び実行オペレーションが行なわれた後、ストアを行うためにデータはLSUデータ・バス210に直接供給される。新しくしてストア・データを得るために一時レジスタ・ファイル252をアクセスする手間が省け、従って命令の実行時間の増大につながる。

LSU205の主な目的は可能な限りCCU110にロード要求をアウト・オブ・オーダで行い、ロード・データをなるべく速くIEU107に返すことである。ロード・オペレーションはアドレス衝突がなく、書き込み実行待ちが存在しない場合のみ実行できる。アドレス衝突が発生するのは、古い命令がまだ書き込み中の記憶場所において読み出しが要求された時である。書き込み実行待ちとは、古い命令が格納オペレーションを要求したがストア・アドレス計算がまだ行なわれていないことである。LSU205はデータ・バス210とアドレス・バス220という二つの部分に分割されている。アドレス・バス220はDAFU230、YMU232、及びCCU110とインターフェースし、

よってアウト・オブ・オーダで実行されるが、ストアは常にイン・オーダで実行される。ロード及びストア用のアドレスは、全てのオペランドが有効で且

つDAFU230がアドレス計算に使用可能になりしたい計算される。DAFU230から物理アドレスを受け取る前にLSU205はキャッシュ要求を行うことができるが、次のクロックサイクルまでに物理アドレスがDAFU230からもYMU115からも来ない場合、キャッシュ要求は取り消される。その場合、キャッシュ要求は後に再発行されねばならない。

各命令パケット当たり1個のアドレスのみが必要で、そのアドレスはロード・アドレスとストア・アドレスの両方として機能する。各命令パケット当たり、2個の32ビット・アドレスはアドレス・バッファ310～313のうち一つに格納される。すなわち、アクセスの最初のバイトが一つのバッファに格納され、アクセスの最後のバイトが別のバッファに格納される。下位12ビットがDAFU130で準備されると、これらのビットは一時バッファ305にラッピングされる。上位20ビットがYMU115で準備されると、次のサイクルで全ての32ビットは適切なアドレス・バッファにラッピングされる（すなわち、Address 1又はAddress 2）。アドレス計算は命令の順序で行なわれず、レジスタ依存性が解消した時行なわれる。アドレス変換の後、有効ビット（図示せず）が命令のアドレス・バッファ310～313に設定され、アドレスが有効であることを示す。両方のアドレスは二つの理由で保持される。すなわち、アドレス衝突の検出とページ交換用のキャッシュ要求である。

IPU106によって使用されるアドレスが仮想アドレスであるのに対し、LSU205によって使用されるアドレスは物理アドレスである。IPU106は、CCU110とYMU115間の調整によって物理アドレスが生成されるに依存しつつ、仮想アドレスに対して動作するのに対し、IEU107ではLSU205が物理アドレス・モードで直接動作することが必要である。この条件が必要である理由はオーバラップする物理アドレスのデータ・ロード及びストア・オペレ

な複雑なオペレーションが可能となる。

- a) R1<-R1+R2+ (R3\*2) H3
- b) [R2]<-R2 OR R4

但し、[x]はアドレスxに存在するメモリ・オペランドである。好適な実施例に於ける命令デコード・ユニット（図示せず）はこれらのCISC型命令を次のようにRISC型シーケンスに分解する。

- a) LOAD[R2+ (R3\*2) +3]>Temp Register  
Execute R1:Temp Register ->R1
- b) LOAD[R2]>Temp Register  
Execute Temp OR R4>Temp Register  
STORE Temp Register to address[R2]

このどちらの場合でも、DAFU230はメモリ・オペランドのアドレスを計算するが、ロード及びストアは同じアドレスを共有しているので1個の命令パケット当たり一つのアドレス計算が必要である。CISC型命令をRISC型命令にデコーディングすることについての説明に関しては1992年3月31日出願の米国特許出願番号07/857,599（代理人登録番号SP032）「CISC型からRISC型命令への変換のためのアライメント並びにデコーディング」（CISC to RISC Instruction Translation And Alignment and Decoding）を参照されたい。当該出願の開示を参照することによって本出願に含まれているものとする。第3回にLSU205のアドレス・バス220の詳細なブロック図を示す。ロード命令は命令ウィンドウから発行され、IEU107に

ーションを伴う、アウト・オブ・オーダで実行される命令が存在する場合、データの保全を保証するためである。データ保全を保証するために、データがストア

命令によって供給された場合、LSU205はストア命令がIEU107によって退避されるまでそのデータをバッファリングする。従って、LSU205によってバッファリングされたストア・データはLSU205にのみ一意的に存在する。同一の物理アドレスを実行済みではあるが未だ退避されていないストア命令として参照する複数のロード命令は、ストア命令が実際には退避されるまで遅らされる。その時点で、ストア・データはLSU205によってCCU110に転送可能となり、次に、CCUのデータ・ロード・オペレーションの実行によって直ちに再びロードされる。

上述の如く、DAFU230によるアドレス計算は1クロック・サイクルで起こり、VNU132によるアドレス変換は次のクロック・サイクルで起こる。アドレスがロード用のアドレスであるならば、キャッシュ要求が行われる。一方、アドレスがストア用のアドレスであるならば、格納を行う前にLSU205は退避信号が送られて来るのを待つ。ロード要求はCCU110に対してアドレス計算の最初のサイクルでも行なえる。この時点で該アドレスの下位12ビットがCCU110に送られ、上位20ビット(ページ番号を含む)はアドレス変換の後、次のサイクルでCCU110に送られる。ロード・ストア・アドレス・バス220が使用可能な場合、該300を介してイミディエト要求をキャッシュ110に対して行なうことができる。現在、ロード・ストア・アドレス・バス220には実行待ちのロード及び/又はストア・アドレスは存在しないので、アドレス衝突の可能性も書き込み実行待ちの可能性も全く存在しない。従って、直ちにキャッシュ110に対して要求を行なえる。

ブロック340には複数のマルチプレクサが含まれているが、このブロックはアドレス・バッファ310～313からキャッシュ要求用のアドレスを選択するために使用される。

LSU205はキャッシュ110に対して要求を行なうためにアドレス・バッファ310～313(即ち、予約ステーション)を使用する。4個のアドレス・バッファ310～313(予約ステーションとも呼ばれる)は中央命令ウィンドウ(図示せず)に含まれる4個のバケットに対応する。IEU107がデコード・ユニット(図示せず)から新しいバケットを要求すると、アドレス・バッファ310～313のうち一つが予約される。アドレス・バッファ310～313は命令番号に従って割り当てられる。最も若い(

オペレーションの内最大のオペランドがマスキング制御のために使用される。各アドレスから、0～4個の最下位ビットがマスクされる。その際、回路400はアドレス・バッファ410～413の各バッファごとに1回、つまり合計4回書き写される(第4図にアドレス・バッファ310のアドレス衝突検出ブロックを示す)。

各ロードの最初のアドレス405、406がアドレス407～418のうち1対おきに比較される。2個の比較の結果とその有効ビット419～424間でANDがとられ、その後全論でORがとられ、その結果アドレス・マッチ450a、450b、450cが生成される。アドレス・マッチ450は次に命令番号比較425～427及びストア・ビット431～433とANDがとられ、その結果衝突チェック450a、450b、450cが生成される。命令番号比較425～427は二つの命令間の比較的

新しさを示す。例えば、命令番号比較425はアドレス・バッファ310中の最初の命令と、アドレス・バッファ311中の最後の命令との間の比較的新しさを示す。第2命令が第1命令よりも古ければ、衝突は存在しない、これら3個の衝突検査はORがとられて、検査を受けている特定のロードのアドレス衝突信号460を生成する。

アドレス衝突の検出においては、各ロードの開始(第1)アドレスが各ストアの第1及び第2アドレスと比較される。一つのロード又はストア・オペレーションは1～10バイトまでのどこかをアクセスするので、衝突が検出されることを保証するためにそれらのアドレスのうち幾つかがマスクされる。このようなマスキングは信号470～475で行なわれる。二つのアドレスが相互に比較される前に、最下位ビットのうちビット0、2、3、又は4がマスクされる。マスクされたアドレスが全く一致する場合(等しい比較)、アドレス衝突の可能性がある。マスクされるビットの数(0、2、3、4)はアドレスが比較されているその二つのアドレスのオペランドのサイズ、そして第1アドレスの最下位の2ビットによって異なる。第1アドレスの最下位2ビットが使用されるのは、間違って検出される衝突の数を制限するためである。マスキングにおいて、最大のオペランド・サイズは次のように使用される。

最新の)命令を示すための履歴ポインターが更新される。この時点において、命令がロード、ストア、その両方を伴うものであるか、あるいはそのどちらも伴わないものであるかが判明する。また、ロード及び/又はストアオペレーションを使用されるデータのサイズも判明する。対応する命令がIEU107によって退避された時に、アドレス・バッファ310～313は割り当て解除される。割り当て解除の後、新しい命令バケットがデコードユニット(図示せず)から受け取られる。ロード・バイパス及びアウト・オブ・オーダー・ロード実行を使用するためには、ストアに対するロード依存性(アドレス衝突)を検出する必要がある。コード依存性はアドレス衝突または実行待ちのストア・アドレスのよって示される。ロード依存性が発生するのは、古い命令がストア・オペレーションを要求した記憶場所と同じ記憶場所でロード・オペレーションが要求された時である。アドレス衝突の検出には、ロードの最初のアドレスが各先行ストアの2倍のアドレス(最初と最後)と比較される必要がある。アドレスの最後のビットとのこのような比較が必要なのは、ストアが4ワード・ページ境界を越えたり、位置合わせがなされていかなかったりするからである。アドレス・ビットのマスキングは偽の

依存性検出を最低限に抑えるためにデータのサイズに応じて行なわれる。ロード・データが4ワード(64ビット)境界からはみだすと、好適な実施例ではそのロード・データにはロード依存性があると仮定される。その理由は、ロードの第2アドレスを各ストアの二つのアドレスと比較するコンパレータが存在しないからである。一つの衝突が検出されると、ロード・オペレーションはその衝突しているストア・オペレーションがCCU110に送られるまで待たなければならない。実行待ちのストア・アドレスとはストアのアドレスが未だ有効でないということを意味する。従って、そのアドレスが判明するまでロード依存性の存在が仮定されるのである。

第4図に、LSU205によって使用されるアドレス衝突ブロック400の概略図を示す。アドレス比較論理による二つのアドレスの比較は最下位ビットのビット0～4がマスクされた後行なわれる。マスキングの後、アドレスが全く一致するならば、これら二つのアドレスの間に衝突が存在することになる。各比較ごとに、二つの

オペランド・サイズ	マスクするビット数
1バイト	0ビット・マスク
2バイト	アドレスが0で終わる場合、1ビットマスク
	アドレスが01で終わる場合、2ビットマスク
	アドレスが11で終わる場合、3ビットマスク
4バイト	アドレスが00で終わる場合、2ビットマスク
	アドレスが11で終わる場合、3ビットマスク
8バイト	3ビットマスク
10バイト	4ビットマスク

更に、ロード・オペレーションが4ワード境界を越えるたびに、アドレス衝突が発生していることが仮定される。その理由は、ロードの最初のアドレスだけがストア・アドレスと比較されるので、アドレス衝突が検出されないことがあるからである。ハードウェア中で使用されるコンパレータの数を二倍に増やすことにより、この制約は削除できる。ストア・アドレスが4ワード境界を越えることがあれば、アドレス衝突は検出される。

マスキングの必要性を次の幾つかの例で示す。(下記の全ての数字は二進数である)。ロードのアドレス2は衝突検査の目的で使用されないので、アドレス2は省略する。

例1:

オペレーション	アドレス1	アドレス2	サイズ	マスク
LOAD	...1001	-	2バイト	2ビット
STORE	...1000	..1011	4バイト	2ビット

ロード・アドレス1001がマスクなしに1000及び1011と比較された場合、ストアがバイト1000、1001、1010、及び1011に書き込んだとしても衝突は検出されない。2個のLSBがマスクされていれば、結果は次のようになる。

オペレーション	アドレス1	アドレス2
LOAD	-1000	-
STORE	-1000	..0000

例2:

オペレーション	アドレス1	アドレス2	サイズ	マスク
LOAD	..0100	-	4バイト	2ビット
STORE	..0001	..1000	8バイト	1ビット

3個の LSB がマスクされていれば、下記のアドレスが生成され、アドレス衝突が検出される。

オペレーション	アドレス1	アドレス2
LOAD	..0000	-
STORE	..0000	..1000

2個の LSB だけがマスクされているならば、下記のアドレスが生成され、アドレス衝突は検出されない。

オペレーション	アドレス1	アドレス2
LOAD	..0100	-
STORE	..0000	..1000

前述の如く、LSU205はキャッシュ要求を必要とする最大4個のロード命令と最大4個のストア命令のウィンドウから選択を行なうことができる。これらのロード及びストアはCCU110に対して互いに競合し、競合するロード及びストア間の選択は下記の如く行なわれる。

ストア命令は、単に他のロード及びストアだけではなく、全ての命令に対してプログラム順序で行なわなければならぬ。ストア要求はストア命令を追跡する信号がIEU107から送られた段階でCCU110に発行される。この信号は、全ての先行命令が終了し、それらの命令では例外も、或って予測された分歧も無かったことを知らせる。ストア命令をこの信号よりも早く行なうことは不可能である。その

理由は、ストアはマシンの状態を不可逆的に変更するので、例外も分岐も発生しなかったことを確認することが重要であるからである。データ・キャッシュ119の使用の目的では

ストアはロードに優先する。その理由は、ストアの遅延はパケットの退避の遅延をもたらし、従って命令デコード・ユニット（図示せず）からの次のデコード読みパケットの受取を遅らせるからである。

ロードが先行ストアに依存しない限り、ほとんどのロード命令はアウト・オブ・オーダーで発行できる。これの例外は、メモリ・マップされたI/Oからの読み出しのような、副作用を持つロードである。本発明の好適な実施例ではメモリ・マップ入出力（I/O）サブシステムが使用される。ある種のI/Oデバイスは読み出しによってアクセスされるとその状態が変化する。例えば、ある種のPIR0バッファは次のデータ項目に番号を付けて、その結果ある種のデバイス状態レジスタは自動的にクリアされる。このようなシステムにおいては、ロード・バイパスは危険なオペレーションである。或って予測された分岐、或いは例外のために、バイパスされたロードが競合して発行されることがある。そのようにバイパスされたロードがシステム状態を不正に変更するような事態が生じてはならない。

この問題を解決する方法はこれらの人間がイン・オーダーで行なわれるようロード/ストア・ユニットを構成することである。ロード/ストア・ユニットでは、キャッシュ要求で要求されたデータがキャッシュ可能であるか否かにかかわらずキャッシュ110に通知を行なう機能が用意されている。この機能によって、プロセサはこのデータがライト・スルーである、つまりキャッシュ可能である、とキャッシュ110に通知することができ、また直ちにメモリはライト・スルーを行なうべきであると通知する。システムの状態を変更する外部読みだしアクセスはこれらのキャッシュ不能アクセスのサブセットであるが、上記の問題は、このデータはキャッシュ不可能であるとのキャッシュ110への通知に関連して

イン・オーダーの要求を行なうことによって解決される。或って、ロードバイパスを完全に回避するかわりに、プロセサはキャッシュ不可能なロードのバイパスを

防止できる。このようにして、ほとんどのロード・オペレーションが、まれに発生するキャッシュ不可抱なロードでの不正なオペレーションを生成することなく、バイパスを利用できるようになる。このような構成はまたメモリ変更以前に例外が発生しないことを保証するためにも必要である。一つのロードが一つのストアに対して依存性を持たない場合、「ストアのロード・バイパス」が発生する。各ロードはページ・キャッシュ使用不可（page-cache-disable）及びページ・ライト・スルー（page-write-through）という2種のビットと対応している。これらのビットはYEU115又はIEU107から得られるビットである。

ストア・データは二ヶ所のうちの一つから生成される。第1に、それは84ビット整数ストア中に整数データ・バス上でLSU205に直接発行できる。第2の方法は整数及び浮動小数点数ユニットによる結果を監視（スヌーピング）することによって行なわれる。これは通常の「実行後格納」シーケンスをサポートするために行なわれる。このシーケンスでは一つの命令の実行の結果はその命令のストア・データである。そうすることによって、" [R2]([R2]OR R4)" のようなCISC型命令の結果が、その命令が明示的にLSU205に発行されなくても格納されるようになる。

LSU205はサイクルごとに一つの要求だけをCCU110に対して行なうことができ、その場合ストア・オペレーションが優先される。書き込み制御がLSU205に対し、この命令は追跡可能であると通知すると直ちにストア・オペレーションはCCU110に送信される。次の優先度はアドレス・バッファ310～313に有効なアドレスを

待ち、アドレス衝突も実行待ちの書き込みも持たない、最も古いロード・オペレーションに与えられる。命令間の比較的な新しさはバッファの位置とバッファ・ポインタの値で決定される。最後に、DPU230から送信された新しいロードが優先度を持つ。この最後の場合、アドレス衝突及び実行待ち書き込みは要求が行なわれるまで後送されず、そして必要ならばロード要求は取り消される。

時折、キャッシュ・ミスが起こる。ストアの場合、CCU110はこのような事態を処理し、その結果LSU205はキャッシュ・ミスの影響を全然受けずに済む。ロードの場合、LSU205はキャッシュ・ミスについて通知を受け、データが返される前に遅延

延が起こる。LSU205は次にキャッシュ・ミスの発生をIEU107に通知し、その結果このデータを待っている命令は取り消される。

目的のデータがキャッシュ・ライン境界を超えると、ロード・オペレーションに対して2個又は3個のキャッシュ・アクセスが必要になります。これらの要求は通常して行なわれ、1サイクル当たり一つの要求が行なわれる。好適な実施例において、一つのキャッシュ・ラインの幅は8バイトで、000で終了するアドレスに位置合わせされている。3個のキャッシュ要求が必要とされるのはI1で終了するアドレスで始まる80ビット・データの場合だけである。このデータがデータ・キャッシュ119から返される場合、ロード・アライナ550（第5回を参照して下記に説明する）が配置され、このデータのシフトとラッチが行なわれる。

ほとんどのロード/ストア・ユニットはデータが行き先レジスタに入るようそのデータをゼロまたはサインで拡張するが、本発明の好適な実施例では、行き先レジスタの初期値が保持され、その一部のみが変更される。勿論、これは8又は16ビット長の整数

ロード・データの場合のみ意味がある。レジスタの初期の内容はアドレス計算の時点でLSU205に送られ、次にデータ・キャッシュ119からのロード・データは初期値データとマージされる。

第5回にLSU整数データ・バス210の概略図を示す。LSUデータ・バス210はロード及び/又はストア・データをCCU110及びIEU107間で転送する。ロード・オペレーション中に、データは端290を介してデータ・キャッシュ119からLSUデータ・バス210に入り、ストア・オペレーション中には端275、276、277を介してIEU107から入る。データ端275及び276は32ビット・データを書き込みバス270を介して機能ユニット260及び262からLSUデータ・バス210に供給し、端282は有効アドレス又はマージされたデータを供給する。有効データがLSUデータ・バス210へ供給されるのは一つの命令の結果が、そのアドレス・コケーションに存在するデータではなく、アドレスそのものである場合である。ストア・データ端516は64ビット・データをLSUデータ・バス210へ供給する。データはデータ端290または292を介してデータ・キャッシュ119又はIEU107のいずれかにそれぞれ返される。

データ・バッファ520～526は、データ・キャッシュ119への或いはデータ・キャッシュ119からのデータ転送中、ロード及び/又はストア・データを保持するため配置されている。各データ・バッファ520～526及びアドレス・バッファ310～313の間に1対1の対応が存在する（そしてこれらのアドレス・バッファ及び4個の命令パケットとの間に1対1の対応が存在する）。各アドレス・バッファ310～313にはLSUデータ・バス210中に2種の対応するデータ・バッファが存在する。すなわち、整数ロード及び整数ストア・データ（8ビット）520～526に対して一つのデータ・バッファと、浮動小数点ロード及びストアのデータ（10ビット）540～546に対して一つのデータ・バッファである。本発明に於いては、浮動

小数点演算用に一つの別個のLSUデータ・バスが存在する。浮動小数点データ・バッファ540～546の動作は整数データ・バスに関して説明された動作と同一である。一つの命令は整数命令合併は浮動小数点命令のいずれかであるので、この二つのユニットは物理的に接続されていないくとも構わない。以下に、整数データ・バッファ520～526の動作のみを詳しく説明する。

制御線581及び587はデータ・フローをそれぞれマルチブレクサ560及び565を介して制御するために配置されている。又、制御線582及び586はデータ・バッファ520～524、524～528へのデータ・フロー、そしてデータ・バッファ520、522、524、及び526からのデータ・フローを制御するために配置されている。

ロード・オペレーションに於いては、データは線290を介してデータ・キャッシュ119からLSUデータ・バス210に入る。ロード・データはアライン・ブロック550に入り、アライン・ブロックはデータの位置合わせを行ない（下記の説明を参照されたい）、位置合わせされたロード・データをマルチブレクサ530～536に転送する。位置合わせされたロード・データは次に、どの命令がデータを要求したかにより、データ・バッファ520～526の一つにラッピングされる。ストア・オペレーション中、ストア・データはデータ線275、276、277を介してIEU107からLSUデータ・バス210へ入り、その後、データ・バッファ520～526のうち適切なデータ・バッファにラッピングされる。

ロード及び/又はストア・データのうちのどちらかがデータ・バッファ520～526

にラッピングされると、そのデータは線290を介してデータ・キャッシュ119へ、或いは線292を介してIEUへのいずれかに送られる。4個のデータ・バッファ520～526はデータをマルチブレクサ560、565に供給し、次にこれらのマルチブレクサは

LSUデータ・バス210から転送されるべき適切なデータを選択する。

しばしば、ストアを含む命令の結果は主メモリ260に格納されなければならない。従って、命令の実行の後、その結果はデータ線275、276を介してLSUデータ・バス210に直接書き込まれる（最初に結果をレジスタ・ファイル250に格納するのではなく）。LSUデータ・バス210は命令の退避信号を受け取るまでデータを適切なデータ・バッファ520～526に保持する。

定期的に、一つの特定な命令は一つの行き先レジスタ全体に格納を行わないようになっている。この場合、「マージ・データ」がデータ線292を介してLSUデータ・バス210に供給される。例えば、一つの命令が8ビットだけを行き先レジスタに格納したいが、残りの24ビットをレジスタに保存したい場合、マージ・オペレーションが行なわれる。従って、データ線292は行き先レジスタの初期値（すなわち、マージ・データ）をLSUデータ・バス210に供給する。マージ・データ（すなわち、行き先レジスタの内容）は適切なデータ・バッファ520～526にラッピングされる。次に、新しい（ロード）データが線290(1)を介してキャッシュから戻され、アライン・ブロック550に入る。アライン・ブロック550はデータの位置合わせを行ない、それをマルチブレクサ530～536に供給する。ロード・データは次に、マージ・データを保持している同じデータ・バッファ520～526にラッピングされる。一旦全てのデータがアセンブルされると、それは適宜な記憶場所（すなわち、データ・キャッシュ119又はレジスタ・ファイル250）に転送可能となる。

従来のロード・ストア・ユニットでは普通、アドレスが特定の境界に位置合わせされなければならない。例えば、32ビット・データ・アクセスでは000で終わるアドレスがなければならない。しかしながら、好適な実例のコンピュータ・アーキテクチャによって8、16、32、64、又は80ビット・データの位置合わせされていないアクセスが可能になる。位置合わせされ

タ・アクセスでは000で終わるアドレスがなければならない。しかしながら、好適な実例のコンピュータ・アーキテクチャによって8、16、32、64、又は80ビット・データの位置合わせされていないアクセスが可能になる。位置合わせされ

ていないアドレスを有することは次のような影響を及ぼす。つまり、(1)ストアに対するロード依存性検出のためにさらに別のハードウェアが必要である。(2)データがページ境界を越えるとアドレス変換が2回必要になる。(3)1つのロードに対して複数のキャッシュ・アクセスが必要になる。

CCU110によって返されたロード・データの長さは8ビットであり、それはデータ・バッファ520～526中の適切な位置に位置合せして格納される必要がある。時には、完全なロードができるまでに2又は3のデータ集合が返されねばならない（例えば、二つ以上のキャッシュ・アクセスが必要な時）。更に、これらのデータ集合がアウト・オブ・オーダで返される場合があるので、特別な措置が必要である。

整数データの位置合わせは8倍の8入力マルチブレクサ（8ビット幅）を使用して処理される。各マルチブレクサはデータ要求の1ビットに対応する。CCU110からロードされた8ビットのデータのうちどのデータが適切なデータ・バッファ520～526にラッピングされるべきかを決定するために8ビットの選択線が使用される。更に、データ・バッファ520～526はどのビットが上書きされるべきかを制御するためにビット・イネーブルになる。

第6図にキャッシュ無交差を持つ位置合わせされていない整数ロードの1例を示す。この例では、アドレスXXXXXX5から4ビットのロードが要求されているが、このロード要求はキャッシュ線からはみだすので、その結果2個のロード要求が必要とされる。最初のキャッシュ要求がデータを返した後、データはロード・ア

ライナ550に転送される。ロード・アライナ550は最後の3ビットをビット0までシフトし、その後、最後の3ビットは適切なデータ・バッファ520～526にラッピングされる。データ・バッファの最後のビットはストアによって上書きされない。一旦第2のキャッシュ要求のデータが返されると、図示されているようにキャッシュ線の最初のビットがデータ・バッファの最後のビットにラッピングされる。更に、この例ではキャッシュ線はイン・オーダで返されるが、それはどの順序で返されても構わない。

浮動小数点データ位置合わせは整数位置合わせと同じ働きをするが、浮動小数点データ位置合わせの場合、10個の8入力マルチブレクサが使用される。

LSU205ではロード・フォワーディングはサポートされていない。ロードがストアに依存する場合、そのロードはロード要求を行なう前に、ストア・データがキャッシュに書き込まれるまで待たなければならない。しかし、本発明の設計では、本質的にロード・フォワーディング機能の実現を阻止するような制約はない。当業者にとって、ロード・フォワーディングを実現するために必要なハードウェア変更を行なうことは容易であろう。

LSU205の好適な実例では多重処理環境がサポートされている。各命令はロード及び/又はストア以外に、ロック或いはアンロック・コマンドを含むことができる。これらの信号はキャッシュに送られ、キャッシュはデータをロックし、メモリ及び入出力サブシステムに同じことをするように通知を送る。ロック又はアンロック・コマンドが命令ウィンドウに存在する場合、ロードはそれらの命令の順序と同じ順序で行なわなければならない。すなわち、後続のロードは最初にロック/アンロック・コマンドを伴うロードを先ず行なわないと実行できない。

#### LSU205のオペレーション例

図8にLSU205のオペレーションを示すサンプル・プログラムを示す。プログラムはインテル486（Intel 486）の表記法で記述されている。3個のレジスタが使用され、それらはeax、ebx、そしてecxとラベルされている。ロードされ、ロード及び/又はストアされるデータは32ビット幅のデータであると仮定される。ブレケットにアドレス・ロケーションを示す。

- (1) mov ebx, [ecx]
- (2) dec ebx
- (3) or [eax], ebx
- (4) (size\_16) mov ebx, [eax+3]

このコードの最初の行では、アドレスeaxに格納されたデータがebxに移される。

従って、この命令は一つのロード・オペレーションである。第2の命令ではレジスタ ebx にある値が減少され、この命令ではロードもストアも行なわれない。第3の命令はアドレス eax に格納されたデータ及びデータ ebx に対して論理和をとり、結果を [eax] に格納する。従って、このオペレーションではロードとストアの両方が行なわれる。最後に第4命令ではアドレス eax+3 に格納された16ビットのデータが ebx に移動される。従って、この命令ではロード・オペレーションが行なわれる。

このコードが実行される前に、下記の値（全て16進法で表記）がレジスタ及びメモリに含まれていると仮定する。

表B

```

eax=0000_0100 [0100]=0000_4321
[0104]=FFFF_FFFF
ecx=0000_1201 [1200]=6500_01FF
[1204]=FFFF_FFB7

```

表Aの命令の実行の結果を表Cに示す。

表C

```

mov ebx,[ecx]    EBX <- [1201]=8765_0001
dec ebx         EBX <- 8765_0001-1 = 8765_0000
or [eax],ebx    [EAX] <- 0000_4321 or 8765_0000=8765_4321
(SEZNE_10) mov ebx,[eax+3] EBX <- [0100+3]=[0103]=FF87>8765_FFB7

```

次に、表Aの命令の実行の結果の詳細を説明する。

第7(a)図から第7(h)図において、LSU205の代表的な例を示す。各図は一つのサイクルを表わす（例えば、第7(a)はサイクル1を表わし、第7(b)図はサイクル2を表わす）。4個のアドレス・バッファ 310～313、及びロード701、ストア702、及び有効ビット717が図示されている。更に、衝突ビット710、実行待ちビット715、及び要求されたデータのサイズ指定705が図示されている。アドレス0100から0

107及びアドレス1200から1207のカレント・メモリ内容は参照番号780として示されている。ブロック730にカレント・キャッシュ要求を示す。ブロック740はデータが（そのようなデータが存在するならば） CCU110から最近返されたことを示す。ブロック760はVMU115から返されているアドレスを示し、ブロック770は DAFU230 0から返されているアドレスを示す。

コード及びストア・ビットはイン・オーダで設定されるが、各アドレスはLSU205にアウト・オブ・オーダで供給されても構わない。ブロック750に、返されたデータが如何にして物理的に位置合わせされるかを示す。

第7(a)図において、最初の命令は「or ebx, [eax]」である。最初に、eaxに格納されているデータはLSUアドレス・バス220に転送されなければならない。eaxに格納されているアドレス、つまり1201は DAFU230から一時アドレス・バッファ305に転送されるが、このアドレス全体は必要でない。最初の12ビットと最下位の3ビットが一時バッファ305に転送される。その理由は、上位20ビットは DAFU230 から VMU115に転送されるからである。xorオペレーションにはロードが伴うので、パケット0に於けるロード・ビットは1に設定される。要求されたデータは32ビットである（ブロック705の011によって示されている）。

これはLSU205中のアドレスの第1集合だから、ブロック730に示すように、アドレス情報は識別子(id)と共に直ちにCCU110に送られる。LSU205は識別子に基づいて、返されたデータがどの命令と対応しているかを決定する。LSU205に転送されるためにアドレスがVMU115によって変換されるのをLSU205が持っている間、一時レジスタ305が使用される。

第2命令の「dec ebx」はアドレス・バッファ・キュに入れられる。decオペレーションにはロードもストアも伴わないで、アドレス・バッファ311に対応するロード・ビット701とストア・ビット702の両方が0に設定される。ロードもストアも必要でないで、この命令ではアドレス計算は必要でない。

第7(b)図において、eaxに格納されたアドレスの第1バイトはレ

ジスタ・アドレス1 (address1)に入れられ、そのアドレスの最後のバイトはア

アドレス2 (address2)に入れられる。これらのアドレスは両方とも勿論アドレス・バッファ310中に存在する。両方のレジスタ (address1 及び address2) は有効なアドレスを含んでるので、両方の有効ビットが設定される。address1及びaddress2が異なった時点でアドレス・バッファ310～313にラッチされることも可能である。これが発生するのは、VMU115からの変換を2回必要とするページ・クロシングが起こった時である。

第3の命令は「or [eax], ebx」である。第3命令に関する (IEU107から送られる) 情報はアドレス・バッファ312に対応する適切なブロックに入れられている。OR命令はロード及びストア・オペレーションを必要とするので、両方のビットとも適宜に1に設定されている。要求されたデータの長さは32ビットで、ブロック705に示されている。更に、第3命令に対応するロード/ストアのアドレスはブロック770に示すように DAFU230から供給される。そして、ブロック730に示すように、このデータに対してキャッシュ要求が行なわれる。

更に、第2サイクル中に、第1命令用に要求されたデータはキャッシュから検索され、データ・レジスタ520に格納されている。しかし、ブロック730に示される、返されたデータは位置合わせされていないデータである。CCU120はアドレス1200で始まるデータのブロックを返したが、命令が要求したデータは1201で始まる32ビットのデータである。従って、ブロック750に示すように、データは位置合わせされなければならない。返されたデータは LD\_ALIGN を00000010に設定することによって2ビットシフトされ、最初の32ビットのデータは BYT2\_SEL によって選択される。第7(c)図において、ブロック770に示すように、次のアドレスが DAFU230によってLSU205に供給される。第3命令に対応

するアドレスはアドレス・バッファ312にラッチされる。有効ビット717の両方のビットが設定される。第1命令がそのオペレーションを完了したので（すなわち、データが CCU110から返され、IEU107に送られたので）、今や有効ビットがリセットされている。（パケット番号が4にリセットされているのは例示の目的のためである。好適な実施例に於いては、ポインタが命令の比較的新しさを管理するために使用される）。第3命令では eax に格納されたアドレスの取り出しが必要

である。アドレスが一旦LSU205に入ると、キャッシュ要求が実行可能となる。更に、第4命令に関する情報、すなわち、その命令はコードで要求されているデータの値は16ビットである（010によって示されている）が、アドレス・バッファ313に対応する適切なブロックで示されているように、この情報がIEU107から送られて来ている。しかし、第4命令より古いストア（すなわち、第3命令）が存在する。LSU205はポインタを使用して、どのアドレス・バッファが最も古い命令情報を含んでいるかを決定する。このストアが存在するので、アドレス・バッファ313に対応する書き込み実行待ちビット715が設定される。従って、この場合、第4命令用のキャッシュ要求は生成できない。

ブロック740に示すように、CCU110は第3命令用にデータをLSU205に戻す。要求データはアドレス100で始まっているので、戻されたデータは位置合わせされる必要がない。最初の32ビットだけが BYT2\_SEL で選択され、そしてデータはデータ・バッファ526にラッチされる。

第7(d)図において、第4命令に対応するアドレスはアドレス・バッファ313にラッチされ、対応する有効ビットが設定されている。次に、アドレス衝突オペレーションが行なわれる。第4命令から

の address1 が第3命令の address1 及び address2 と比較され、その結果アドレス衝突の存在が決定される。従って、アドレス・バッファ313に対応する衝突ビット710が設定される。衝突が存在するために、サイクル4の期間中キャッシュ要求は生成できない。しかし、キャッシュ要求の実行が不可能であっても、ブロックで示されているように第4命令用のマージ・データがIEU107から到着する。マージ・データとはレジスタ ebx からのデータである。マージ・データが必要なのは、第4命令が単に16ビット・オペレーションであるからである。このマージ・データはデータ・バッファ526にラッチされる。

第7(e)図において、書き込みデータ (TRA-DATA) がIEU107から到着する。TRA-DATAは第3命令に於けるOR演算の結果である。このデータはデータ・バッファ524にラッチされる。更に、ブロック780に示すように、サイクル5の期間中次のパケット、即ちパケット1、が選択される。具体的には、retire\_nextビットが1

に設定され、次の命令が退避可能であることを示し、retire\_numビットが1に設定され、パケット1にある命令を退避すべきであることを示す。この場合、第3命令と4命令の間にアドレス衝突がまだ存在する。

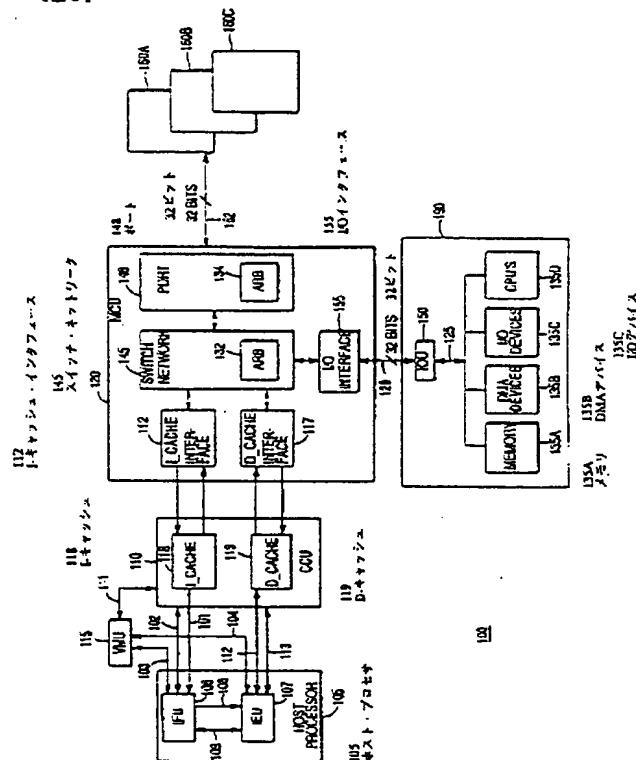
第7(f)図において、データ・バッファ524中のデータはebx中のデータとORがとられ、その結果値87654321が生成される。ブロック785に示すように、ナイクル6の期間中、第3命令が退避される。第3命令の退避によって、LSU205は第4命令に対応する衝突ビット710をリセットできるようになる。ブロック780に示すように、OR演算によって生成された値を記憶場所00000100（レジスタeaxに格納されているアドレス）に格納するためのキャッシュ要求が行なわれる。ブロック780に示すように、データはこのデータ・ロケーションに格納されている。

第7(g)図において、第4命令は記憶場所0103（レジスタeaxの最初の16ビット）に格納されているデータをロードする。従って、ブロック730に示すように、第4命令に対応するロード・オペレーション用にキャッシュ要求が行なわれる。

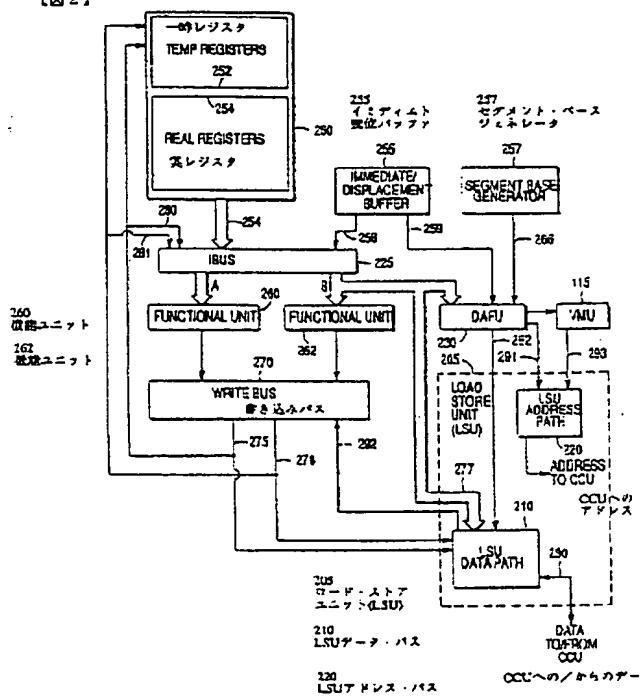
第7(h)図において、ブロック740に示すように、要求された（位置合わせされていない）ロード・データがキャッシュから返される。ブロック750に示すように、次にデータは3バイト左シフトすることによって位置合わせされる。その理由は、要求されたデータはアドレス0100でなくアドレス0103で始まるからである。最初の16ビットだけが要求されたので、最初の2バイトだけが位置合わせされたデータから選択される。これら16ビットは次にデータ・バッファ326にラッピングされ、このデータ・バッファは120107に逆方向に転送されて、レジスタebxに格納される。

上記に本発明を実施例を参照しつつ説明したが、本発明の精神及び特許請求の範囲から逸脱することなく、形状並びに詳細において様々な変更が可能なことが当業者には理解されるであろう。

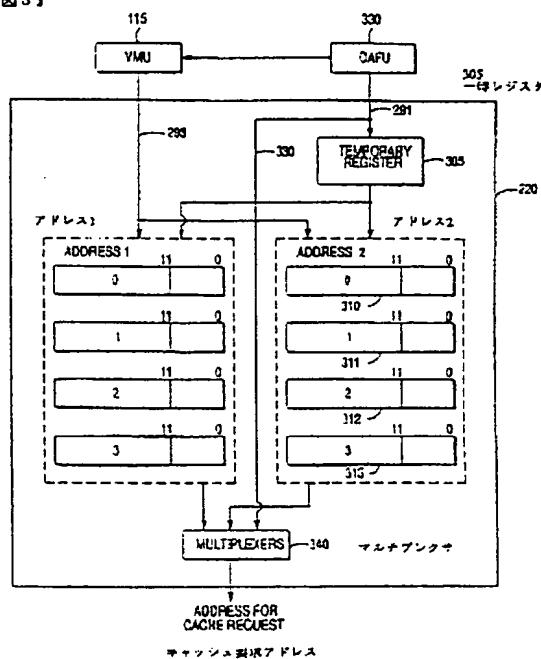
【図1】



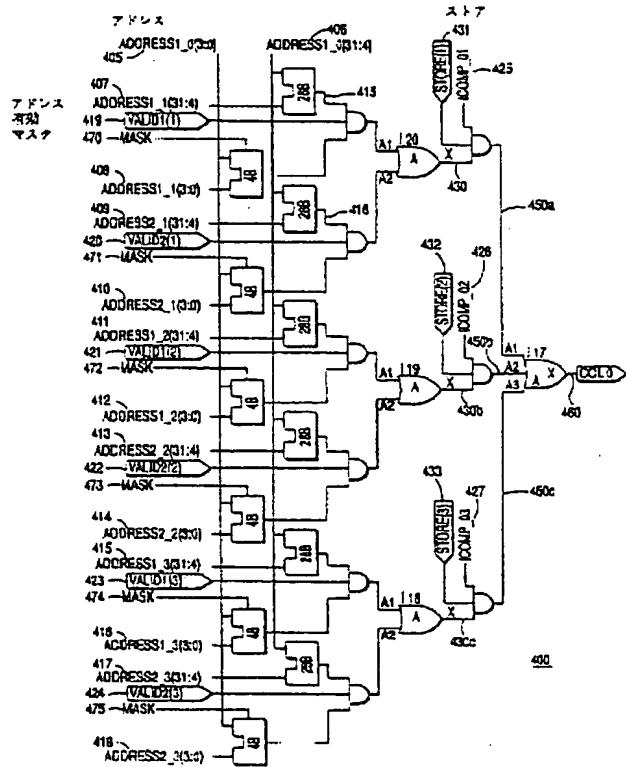
【図2】



【図3】



[図4]



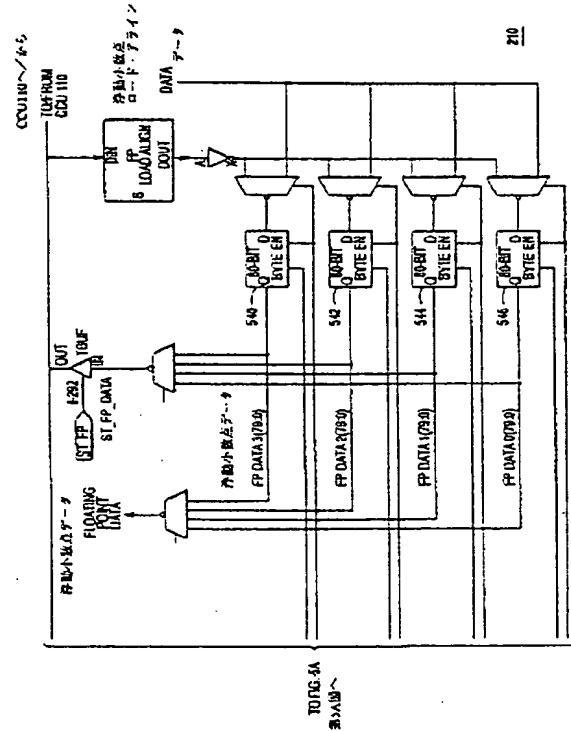
447

特表子3-504977

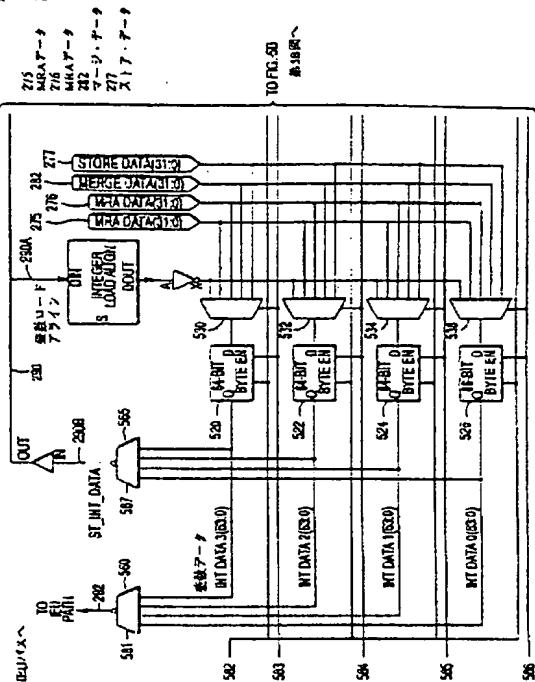
45

特森平3-504977

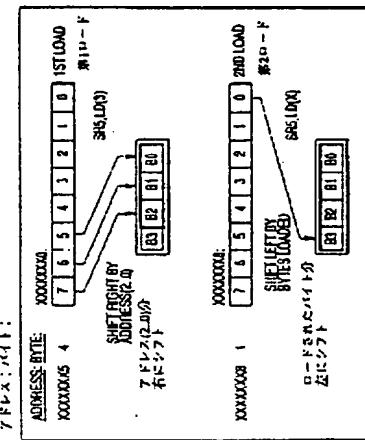
〔四〕 331

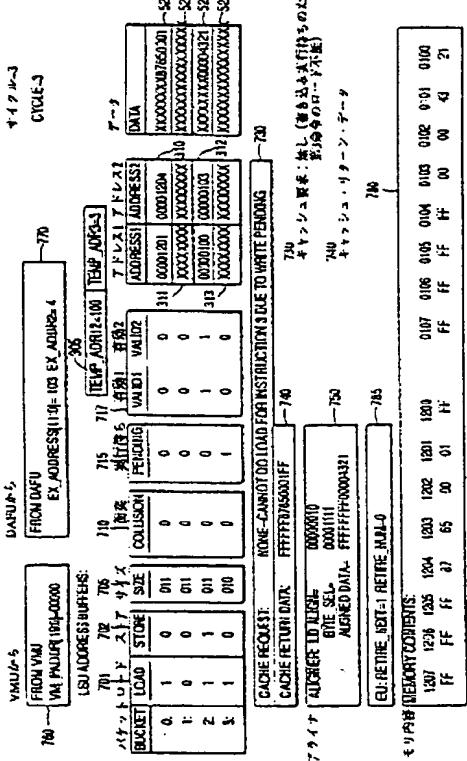


[图5A]



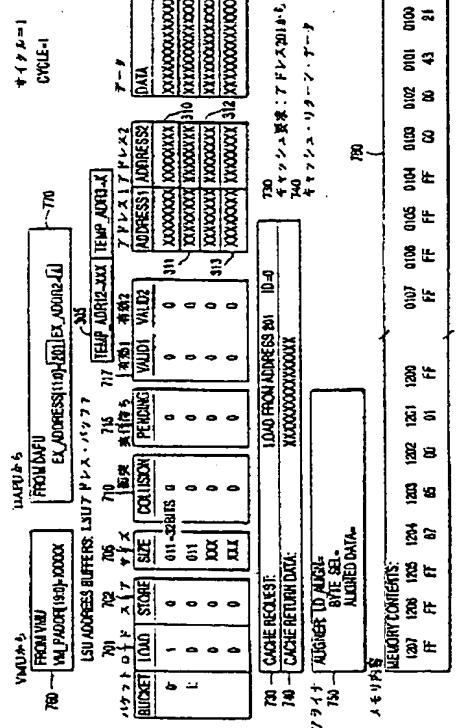
〔四六〕





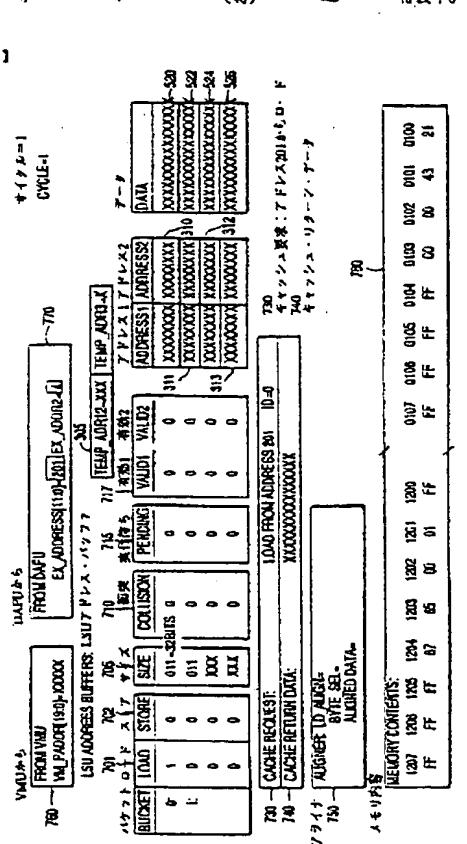
特表平3-504977

特表平8-504977



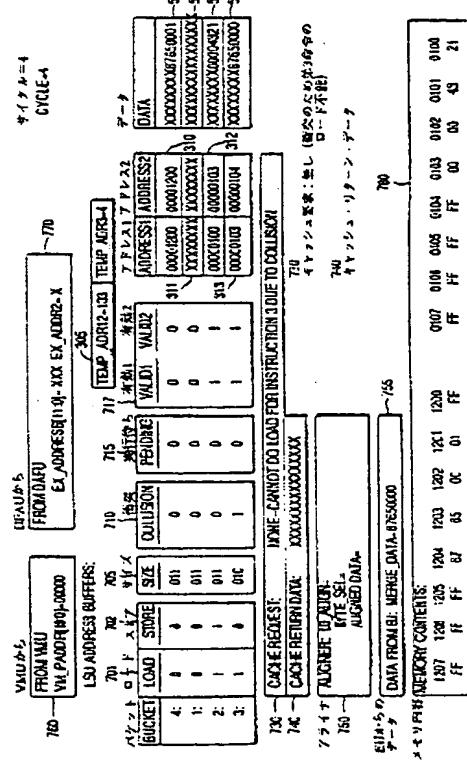
(48)

特志

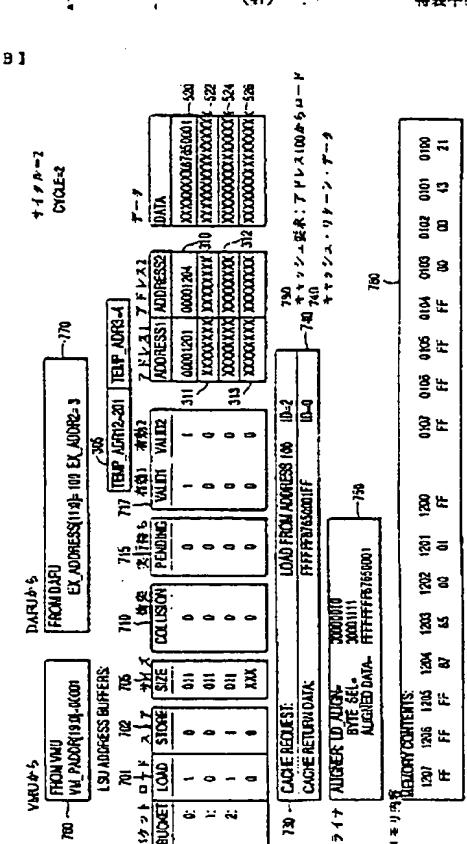


特表平3-504977

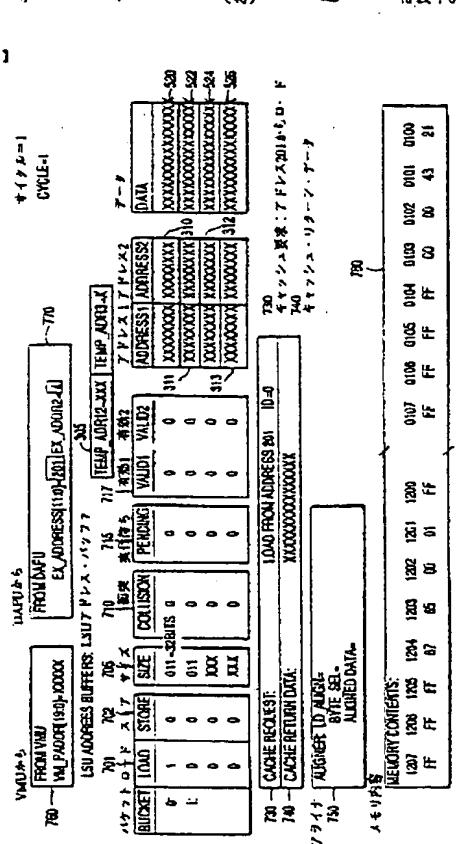
特志



〔圖7D〕

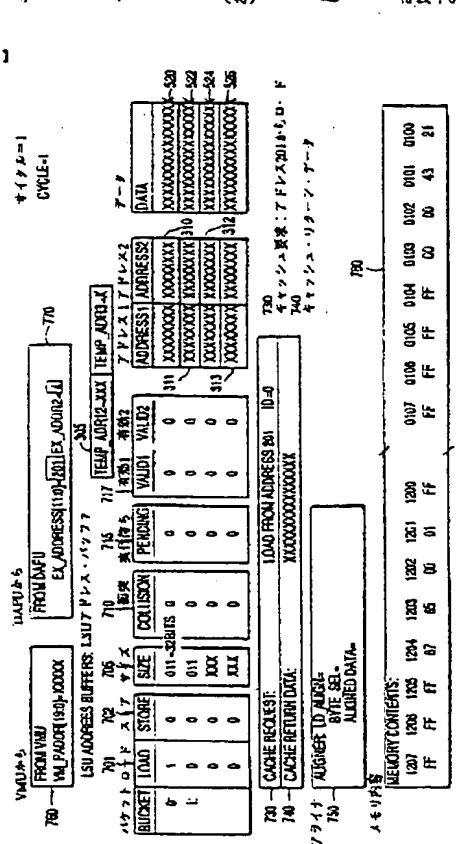


1



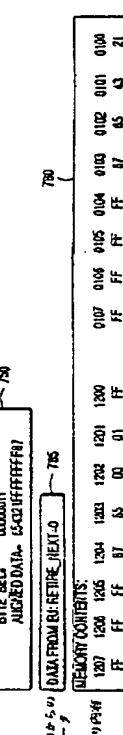
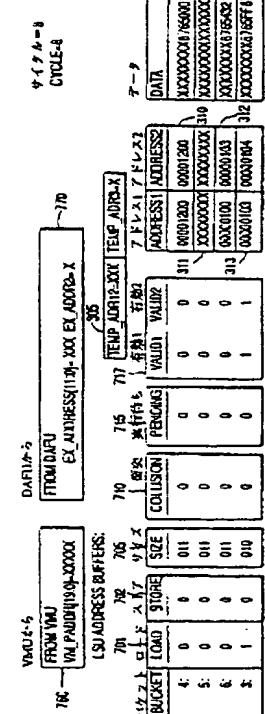
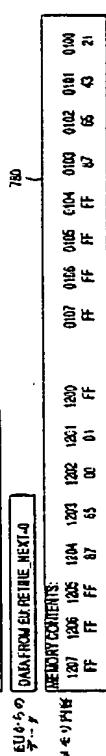
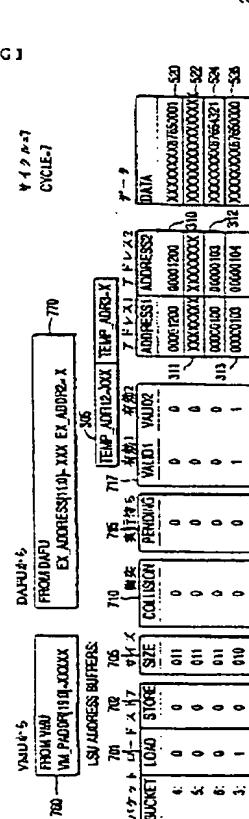
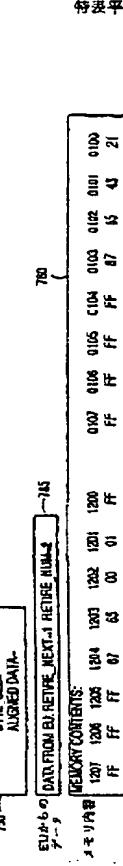
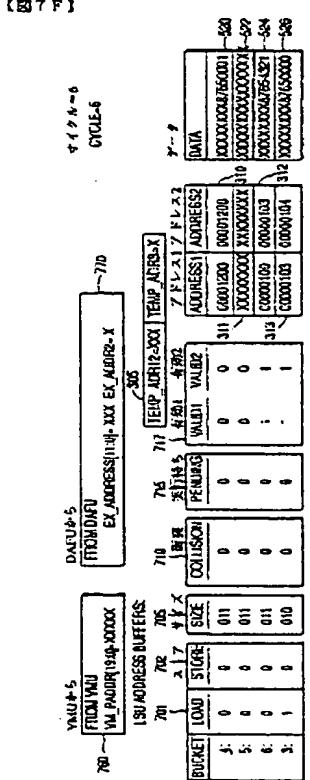
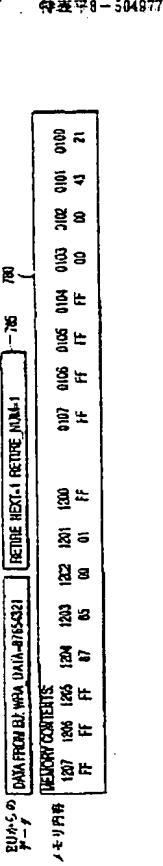
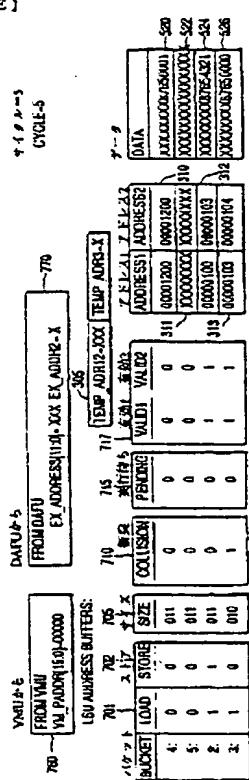
(48)

特志



特表平3-504977

特志



## 【国際調査報告】

## INTERNATIONAL SEARCH REPORT

Date of Application No  
PCT/US 93/08331

A. CLASSIFICATION OF SUBJECT MATTER  
IPC 5 G06F9/38

According to International Patent Classification (IPC) or to both national classification and IPC

## B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)  
IPC 5 G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronics data base consulted during the international search (name of data base and, where practical, search terms used)

## C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	EP,A,0 302 999 (INTERNATIONAL BUSINESS MACHINES CORP.) 15 February 1989 see the whole document	1,2,8,9, 11,13, 16,17, 19,20,22
Y	EP,A,0 147 775 (HITACHI LTD) 10 July 1985 see the whole document	1,2,8,9, 11,13, 16,17, 19,20,22
A	EP,A,0 436 092 (INTERNATIONAL BUSINESS MACHINES CORP.) 10 July 1991 -/-	1,4, 10-12, 16,17, 20,21

Further documents are listed in the continuation of box C.

Patent family members are listed in annex.

## \* Special categories of cited documents:

- 'A' document defining the general state of the art which is not considered to be of particular relevance
- 'B' earlier document but published on or after the international filing date
- 'E' document which may throw doubt on priority claim(s) or which is cited to establish the publication date of another document or other special reason (as specified)
- 'O' document referring to an oral disclosure, etc., exhibition or other means
- 'P' document published prior to the international filing date but later than the priority date claimed

'T' later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

'X' document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

'Y' document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

'Z' document member of the same patent family

Date of the actual completion of the international search

Date of mailing of the international search report

10 December 1993

17.12.93

Name and mailing address of the ISA  
European Patent Office, P.O. Box 8 Patentzaal 2  
NL - 2230 HV Haarlem  
Tel. (+ 31-70) 340-2040, Tx. 31 651 800 NL  
Fax (+ 31-70) 340-3016

Authorized officer

Weinberg, L

## INTERNATIONAL SEARCH REPORT

Int'l. Appl. No.  
PCT/US 93/08331

C. (Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT		
Category	Claim or document with indication, where appropriate, of the relevant passage	Relevant to claim No.
A	DE,A,36 03 240 (SIEMENS) 6 August 1987 see the whole document ----	1-3, 16, 17
A	EP,A,0 259 095 (AMDAHL CDRP) 9 March 1988 see the whole document ----	1, 11, 16, 17
A	EP,A,0 377 990 (INTERNATIONAL BUSINESS MACHINES CORP.) 18 July 1990 see the whole document ----	1, 3, 16
A	IEEE TRANSACTIONS ON COMPUTERS vol. 37, no. 5, May 1988, NEW YORK US pages 562 - 573 SMITH AND PLESZKUN 'Implementing precise interrupts in pipelined processors' see the whole document -----	9, 10, 12, 13, 15, 18-21
A	EP,A,0 272 198 (UNITED TECHNOLOGIES CORP.) 22 June 1988 see the whole document -----	7

## INTERNATIONAL SEARCH REPORT

Information on parent family members

Int. Appl. No.  
PCT/US 93/08331

Parent document cited in search report	Publication date	Parent family member(s)		Publication date
EP-A-0302999	15-02-89	US-A- JP-A-	4991090 63293639	05-02-91 30-11-88
EP-A-0147775	10-07-85	JP-A- US-A-	60129840 4638429	11-07-85 20-01-87
EP-A-0436092	10-07-91	US-A- JP-A-	5185871 3201130	09-02-93 03-09-91
DE-A-3603240	06-08-87	NONE		
EP-A-0259095	09-03-88	AU-B- AU-A- JP-A- US-A- US-A- US-A-	587714 7744387 63113648 4855904 4872111 4722046	24-08-89 03-03-88 18-05-88 08-08-89 03-10-89 26-01-88
EP-A-0377990	18-07-90	US-A- JP-A-	4961162 2227768	02-10-90 10-09-90
EP-A-0272198	22-06-88	CA-A- JP-A- US-A-	1278382 63192135 4992934	27-12-90 09-08-88 12-02-91

---

**【要約の続き】**

ア・ユニットの三つの主なタスクは(1)アウト・オブ・オーダのキャッシング要求の処理、(2)アドレス衝突の検出、及び(3)データの位置合わせ、である。